Pattern Recognition Letters
journal homepage: www.elsevier.com

DFP-ALC: Automatic Video Summarization Using Distinct Frame Patch Index and Appearance based Linear Clustering

Sivapriyaa Kannappan[a,**], Yonghuai Liu[a,b], Bernard Tiddeman[a]

[a]*Department of Computer Science, Aberystwyth University, Ceredigion SY23 3DB, UK*
[b]*Department of Computer Science, Edge Hill University, St Helens Road, Ormskirk, Lancashire L39 4QP, UK*

ABSTRACT

Video summarization aims to create a succinct representation of videos for efficient browsing and retrieval. We propose an innovative method for the task. It includes two main steps: (i) the first step proposes a Distinct Frame Patch (DFP) index for selecting a set of good candidate frames, and (ii) the second step proposes a novel Appearance based Linear Clustering (ALC) to refine them for distinct ones. While the first step measures the content of frames, the second step considers to what extent one frame is different from another in both the spatial and temporal spaces. The experiments are performed over two publicly accessible datasets. The results show the effectiveness and efficiency of the proposed method when compared with other state-of-the-art techniques.

1. Introduction

Due to the recent advances in multimedia technologies and internet services, capturing and uploading videos is so easy that enormous quantities of new videos are available online every second. As stated by YouTube statistics in 2018 [1], around 300 hours of video are uploaded to it every minute. In other words, more than 400 thousand hours of new video are uploaded in a day, which tend to increase the need for suitable video summarization, video browsing, indexing and retrieval in order to manage the prevailing abundant data. Among the multimedia types (such as text, image, graphic, audio and video), video (being a major source of *Big Data*) is the most demanding one as it combines all the other types of multimedia data into a single data stream. It is tricky to gain efficient access to the video, due to its unstructured format and variable length (Zhuang et al., 1998). In this situation, it is essential to develop an automatic means of generating a concise representation of video content called a Video Summary.

According to Troung and Venkatesh (Truong and Venkatesh, 2007), there are two types of video summaries: *Static video summary* (also called *representative frames, still image abstract* or *static storyboard*) and *Dynamic video skimming* (also called *video skim, moving image abstract* or *moving storyboard*). Static video summaries consist of a set of keyframes, whereas dynamic video summaries consist of a set of shots extracted from the original video (De Avila et al., 2011). The major benefit of video skimming is that the summary includes both audio and motion elements, which enrich the emotions and the amount of information delivered by the summary. It is also stimulating to view a skim with an audio-visual component rather than a slide show of static keyframes. On the other hand, static video summaries are not restricted to timing and synchronization issues and are more flexible compared to sequential display of video skims (De Avila et al., 2011). If required, static video summaries can incorporate both spatial and temporal information (key events in a precise order) which assists the user to swiftly grasp the video content (Truong and Venkatesh, 2007). When the static video summarization is performed using clustering techniques, even though the temporal order is usually not maintained, it can still be recovered by automatic ordering of extracted keyframes based on their frame indexes in the original video. Thus, we concentrate on static video summaries in this paper.

Recently, the research in video summarization has gained more interest among the research community and as a result, several techniques have been proposed in the literature incorporating the learned features based on Determinantal Point Pro-

**Corresponding author: Tel.: +44 1970 628762; fax: +44 1970 628536;
e-mail: sik2@aber.ac.uk (Sivapriyaa Kannappan)
[1] https://merchdope.com/youtube-statistics/

cesses (DPPs) (Zhang et al., 2016a), Long-Short Term Memorys (LSTMs) (Zhang et al., 2016b; Mahasseni et al., 2017), viewpoint optimization (Kanehira et al., 2018) and retrospective encoders (Zhang et al., 2018). Though learned features provide better performance, their interpretation, analysis and visualization is very hard, due to their high dimensionality and abstraction. Hence, we attempt to develop an automatic video summarization technique based on hand-crafted features, as most of the prior work concentrated on clustering (Wu et al., 2017; De Avila et al., 2011; Furini et al., 2010; Mundur et al., 2006). The basic idea of clustering is to group similar frames together via shot detection or feature extraction (e.g., color or motion) and then extract a frame (nearer to the cluster center) per cluster as a keyframe. De Avila *et al.* (2011) proposed a method, VSUMM, for producing static video summaries. It is based on color feature extraction from video frames and K-means clustering along with a novel approach for the evaluation of static video summaries. Mundur *et al.* (2006) proposed an automatic clustering algorithm based on Delaunay Triangulation. Furini *et al.* (2010) proposed an approach called STIMO, a summarization technique designed to produce on-the-fly video storyboards. Dang and Radha (2014) introduced a new image feature called Heterogeneity Image Patch (HIP) index. It provides a new entropy-based measure of the heterogeneity of patches using Sum Absolute Difference (SAD). It is evaluated for every frame in a video sequence, in order to generate a HIP curve for that sequence. Subsequently, a set of candidate frames is selected from abundant video frames, based on the HIP curve. Then an Accumulative Patch Matching Image Dissimilarity (APMID) measure is proposed for the creation of an affinity matrix among these candidates. Finally, keyframes are extracted from the affinity matrix using a min-max based algorithm. This method is computationally inefficient in computing the HIP index, as it relies on SAD of pixel intensities between the patches.

Though there are some techniques that produce quality summaries, they are usually computationally expensive and inefficient in a way that the time taken for computing a video summary was around 10 times the video length (Mundur et al., 2006). This is quite inconvenient. In this case, the major challenge still remains in the development of methods for a swift extraction of keyframes for the representation of the content of the full video.

Thus, we propose to improve the HIP index by considering the color histogram feature for each frame in the YCbCr color space with 16, 4 and 4 bins respectively. To reliably determine whether a patch is similar to or different from any existing ones within a class, we propose to represent each class using the average of the features of all the similar patches, leading to an efficient *Distinct Frame Patch (DFP)* index. Initially, the DFP index is used to select a set of good candidate frames. For their refinement, we propose an efficient clustering approach coined as *Appearance based Linear Clustering (ALC)*, in order to generate a static summarization of the original video. Specifically, the selected candidate frames (based on the DFP index) are represented as color histograms and are then clustered based on 2D distance in both the feature space and time space, by searching only the neighboring clusters, until it converges.

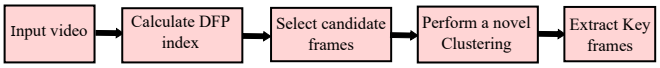

Fig. 1. The main steps in the proposed method.

The rest of this paper is structured as follows: Section 2 details the proposed DFP-ALC method; Section 3 presents the experimental results compared with other state-of-the-art techniques; and finally Section 4 draws conclusions and indicates future work.

2. The Proposed DFP-ALC Keyframe Extraction Method

The proposed method employs a patch-based technique for candidate frame selection. The major motivation lies in two aspects: (i) patches are more resistant to imaging noise than pixels themselves and thus can provide a more reliable representation of how the video content changes spatially; (ii) patches are used to reliably estimate the frame index, then to detect the discrepancy between different frames and finally retrieve the distinct ones. A pipeline of our proposed method is shown in Fig. 1. Initially, each resized frame of a given video is split into different patches, and these patches are then classified into distinct classes to estimate the uniqueness, Distinct Frame Patch (DFP) index, of the frame as the entropy of the classification of these patches whether they are similar in the YCbCr color space. Based on this uniqueness measure, we select a set of good candidate frames. To further refine these frames for the selection of distinct ones, we propose a novel Appearance based Linear Clustering (ALC). The proposed method is elaborated in the following subsections.

2.1. Frame Representation and Selection of Candidate Frames

The detailed description of the method is depicted in Fig. 2 and explained in the following subsections. Pixel intensities are very sensitive to imaging noise and illumination conditions, so how to represent the content of a video is challenging. To address this issue, we propose first resizing each frame and then splitting it into equal sized patches. A color histogram is defined for each patch in a frame and is used to group similar ones into the same class and dissimilar ones into new classes. The process continues until all the patches of a frame are classified into distinct classes based on a preset threshold. Here the frame patches are scanned from left to right and top to bottom. Consequently, the frequency for each distinct class (containing similar frame patches) is calculated, which in turn is used to compute the DFP index and the process continues for all the frames in a video. Eventually, frames having the maximum DFP index are selected from each specified segment of the video as the candidate ones.

2.1.1. Resizing and Patch splitting

Firstly, each video frame in any dataset is down sampled to 80×60 and split into patches with a size of 10×10 pixels. In this case, each frame contains a total of 48 patches. Down

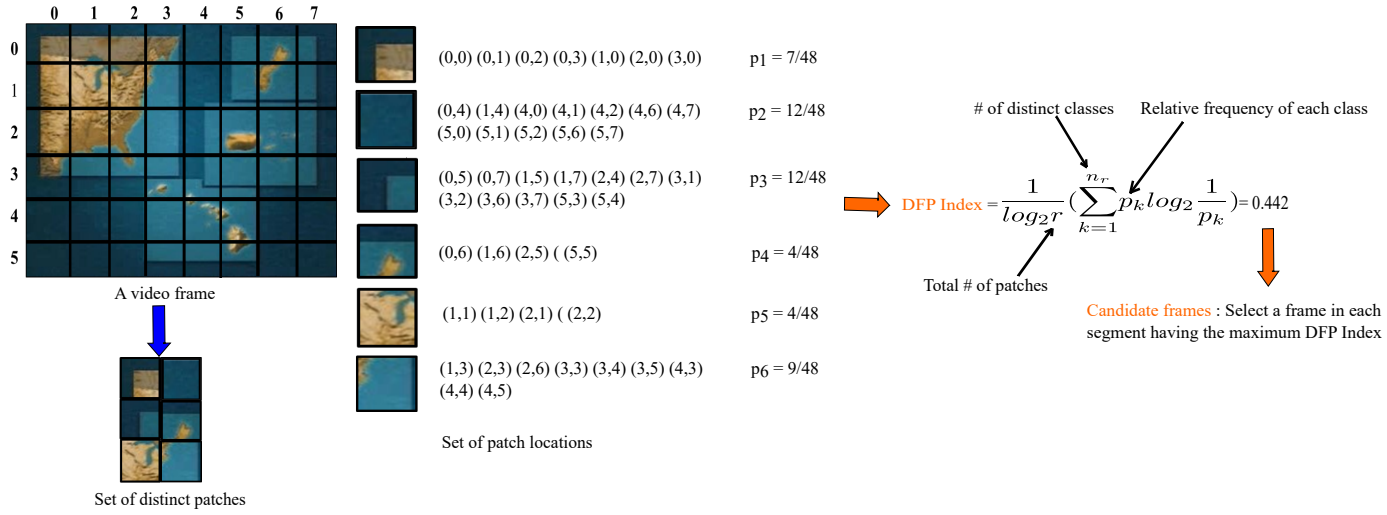

Fig. 2. The estimation of the Distinct Frame Patch Index of a frame in the video *Hurricane Force - A coastal perspective, segment 3* in the Open Video Project dataset.

sampling is performed to reduce the computational time and it is based on the aspect ratio (4:3) of the datasets used in our experiments. The use of different patch sizes usually produces similar results as discussed in Section 3.1.5 and is also noticed in Dang and Radha (2014).

2.1.2. Patch Representation

Formally, a video frame is represented using two sets U and L_U where U denotes the set of different patches and L_U denotes their locations in a frame. Each patch is represented as a feature vector $u_i \in \mathbb{R}^d (i \in [1, r])$ in the d dimensional space (where d=16 $\times$ 4 $\times$ 4 = 256 bins), r is the total number of patches obtained in each frame. Therefore, U and L_U can be represented as (Dang and Radha, 2014):

$$U = \{u_i | u_i \in \mathbb{R}^d, i \in [1, r]\}, \tag{1}$$

$$L_U = \{l_{u_i} \in \mathbb{N}^2 | u_i \in U, i \in [1, r]\}. \tag{2}$$

Subsequently, each patch in a frame is represented using a three-dimensional histogram H in the YCbCr color space with 16, 4 and 4 bins for luminance and chrominance respectively. To identify the distinct patches within the frame without being affected by the lighting conditions, the YCbCr color space is chosen.

2.1.3. Patch Classification

After each patch u_i in a frame U has been represented with a histogram H_i, the Euclidean distance between the histograms of different patches is calculated in order to identify the distinct ones. If the distance between the histograms H_i and H_j of two patches u_i and u_j is smaller than a threshold ϵ: $\|H_i - H_j\| < \epsilon$, then u_i and u_j belong to the same class. In this case, a class will contain more than one patch. If a class contains more than one patch, then the average of the histograms of the patches is computed for their representation and is used for further comparison of the incoming patches, where the ordering of the patches within a class doesn't matter. Otherwise, if $\|H_i - H_j\| \geq \epsilon$, then u_j is distinct from u_i and will be assigned to a new class. The distinct patch classification threshold, $\epsilon = 0.001$, is selected as the maximum threshold based on the *Fingerprint* image specified in Dang and Radha (2014) that leads to a DFP index value of 1. The relative frequency of different patch classes U in a frame for a given threshold ϵ is given by:

$$P_U(\epsilon) = \{\bar{u_k}, p_k | \bar{u_k} \in U, k \in [1, n_r]\} \tag{3}$$

where

$$\sum_{k=1}^{n_r} p_k = 1 \text{ and } 0 < p_k \leq 1 \tag{4}$$

$$p_k = \frac{|[\bar{u_k}]|}{r} = \frac{\#\ of\ similar\ patches\ in\ class\ k}{Total\ \#\ of\ patches} \tag{5}$$

where p_k is the relative frequency of class k, n_r is the number of distinct classes, $|[\bar{u_k}]|$ is the cardinality of the set $[\bar{u_k}]$ which contains similar patches in class k, and r is the total number of patches.

2.1.4. Distinct Frame Retrieval

The DFP index D_U of different patch classes U in a frame is defined as the normalized entropy of $P_U(\epsilon)$ and is given by:

$$D_U = \frac{1}{log_2 r}(\sum_{k=1}^{n_r} p_k log_2 \frac{1}{p_k}). \tag{6}$$

It measures how the patches differ from each other and thus the content of a frame: The more dissimilar the patches, the larger the DFP index. When all the patches are completely different from each other, it will reach the maximum value of 1. After calculating the DFP index for each frame in a video, a frame with the maximum DFP index (highest entropy) from each specified segment is selected as a candidate one for keyframe retrieval. The size of the segment is detailed below in Subsections 3.1 and 3.2 for two different datasets respectively.

2.2. Keyframe Extraction

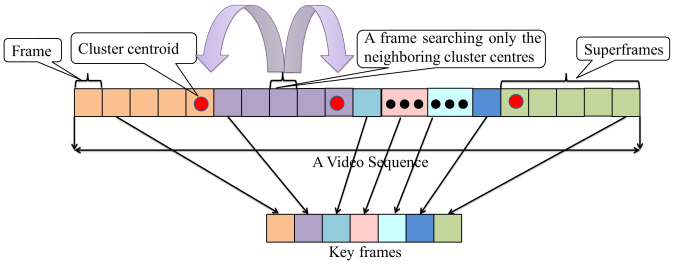

Fig. 3. The main idea of the Appearance based Linear Clustering.

Inspired from SLIC superpixels (Achanta et al., 2012), in this section, we propose a superframe based clustering method in order to extract keyframes from the set of candidates obtained from the previous step. In particular, a group of similar frames can be treated as a superframe. Color is one of the most common features perceived by humans and it is widely used as a visual feature in image processing applications. Color histogram is a common way to represent how the intensities of different pixels distribute. The use of some prior knowledge of the video content help to select appropriate color space for their representation. To this end, since the scenes tend to change rapidly in the open video dataset, Hue alone in the HSV color space is chosen for clustering the candidate frames. However, the scenes are very similar in the consumer video dataset and the YCbCr color space is sensitive to lighting conditions, we choose it for clustering the candidate frames for that dataset. In line with the previous work used for comparison, the number k of clusters is set equal to the number of keyframes in the ground truth for the consumer video dataset. In contrast, for the open video dataset, we propose to use Bayesian Information Criterion (BIC) (Vermaak et al., 2002) for its determination as detailed in the following subsection.

2.2.1. Bayesian Information Criterion

The Bayesian Information Criterion (BIC) is defined as:

$$BIC = \frac{1}{\sigma^2}\sum_{i=1}^{k}\sum_{t\in I_i} d^2(f^{(i)}, f_t) + k\log N \tag{7}$$

where the parameter σ is set to reflect the width of the clusters and experimentally determined as $\sigma = 0.6$, $k \in [2, N]$, I_i is the set of indexes of the frames in a given video that belong to cluster i and $t \in I_i$ is the index of a frame. $d^2(f^{(i)}, f_t)$ is the squared distance between the color histogram features $C^{(i)}$ and C_t of the cluster i and the individual frame t in that cluster, and N is the number of candidate frames. The definition of BIC can be abbreviated as:

$$BIC = measure\ of\ fit + penalty \tag{8}$$

where the first term measures the extent to which the frames in a cluster are different from each other, the second term regularizes the number k of the clusters to avoid the extreme case that each frame forms a cluster. The former decreases with the increase in the latter and vice versa. Thus BIC can be minimized for the optimization of k.

2.2.2. Appearance based Linear Clustering

The selected candidate frames from the last subsection are clustered for the selection of the final keyframes. To this end, the clusters need to be initialized. We propose firstly to split the ordered candidate frames into k segments of equal length $S = N/k$, where N is the total number of retrieved candidate frames and k is the optimal number of clusters. For example, if $N = 100$ and $k = 4$, then the first 25 frames are allocated to the first cluster, the next 25 frames to the second and so on. The process of initializing the centroids in a sequential order at regular intervals will make sure that the consecutive similar frames are in the same cluster and thus would enable faster convergence.

The candidate frames (without any resizing) are clustered based on 2D distance metric in the *feature space* (color space as discussed in Sections 3.1 and 3.2) and *time space* (frame index). Each frame is assigned to either the preceeding or subsequent cluster as shown in Fig. 3. The rationale for this search strategy are two fold: (i) speeding up our clustering process, since the search of a frame for a potential cluster is limited to its local neighbors and thus reduces the number of distance calculations, and (ii) making sure that similar frames at different time intervals can be retained to accurately represent the occurrence of events without ambiguity. Such strategy is in contrast with the conventional k-means clustering, in which each frame must be compared to all the clusters. A similar technique was adopted in (Achanta et al., 2012). The 2D distance metric is defined as:

$$D = d_f + (\frac{d_t}{S}) * \alpha \tag{9}$$

where d_f and d_t represent the distance in the feature space and time space respectively, the parameter α controls the relative importance of the two spaces and is set to 0.5 based on the experimental tests. d_f is calculated as the distance between the feature C_t of the current frame t and that $C^{(i)}$ of a neighbouring cluster i: $d_f = \|C_t - C^{(i)}\|$. d_t is calculated as the distance between the frame index $t^{(i)}$ of the cluster centre i and that t of the current frame: $d_t = |t^{(i)} - t|$.

After clustering the candidate frames, the cluster centroids and indexes are updated by recalculating the averages of the color histograms and frame indexes of the frames in different clusters, until convergence. Finally, the candidate frames closest to the cluster centroids are retrieved as the final keyframes. It can be seen from the development that the proposed clustering method is simple, eliminates redundancy and converges rapidly, thereby increasing the computational efficiency. It has a computational complexity of $O(M)$ in identifying the candidate frames, $O(2N)$ in clustering these candidate frames, and $O(kN)$ in retrieving the final keyframes. Thus it has an overall linear computational complexity in the number M of frames in the original video.

3. Experimental Results

In this section, we experimentally validate the proposed DFP-ALC method. According to Troung and Venkatesh (Truong and Venkatesh, 2007), the existing video summary

evaluation techniques can be grouped into three main categories: (i) result description, (ii) objective metrics and (iii) subjective metrics or user studies. Among those techniques, we selected the subjective metric (quantitative comparison) and result description (qualitative comparison) approaches. The performance of the proposed technique is measured as the number of matches between the automatically selected keyframes and those from the ground truth. In the case of the open video database (De Avila et al., 2011), the ground truth was built by a number of users from the sampled frames over each video clip. In the case of the consumer video database (Luo et al., 2009), the ground truth was identified as those agreed by multiple human judges. For the proposed method, unless otherwise stated, the parameters were set as discussed in Section 2. All experiments were carried out on a computer with a processor of Intel core i7, 3.60 GHz and a RAM of 8GB.

3.1. The Open Video Dataset

In this section, we use 50 videos selected from the Open Video Project (OV) to validate the proposed technique. Those videos are in MPEG-1 format containing 30 fps with a resolution of 352×240 pixels. The videos are distributed among several genres (documentary, ephemeral, historical, and lecture) and their duration varies from 1 to 4 min. We compare our proposed DFP-ALC method with three other keyframe extraction approaches including RPCA-KFE (Robust Principal Component Analysis Key Frame Extraction) (Dang and Radha, 2015), STIMO (STIll and MOving Video Storyboard) (Furini et al., 2010) and DT (Delaunay Triangulation) (Mundur et al., 2006). To initialize the clustering process in our method, we split the video such that 1 frame was selected in every 5 frames for the first 25 frames, then 1 frame in every 25 frames and finally 1 frame in every 5 frames in the last 25 frames, since the beginning and the end parts of the video tend to play a more important role in describing the events in the video. The selected candidate frame t is represented as a color histogram C_t with Hue alone in the HSV color space with 16 bins. To eliminate similar keyframes obtained, we compare them through the color histogram (Hue alone in the HSV colorspace with 16 bins) based on the Chi-square distance. If the histogram difference between the retrieved neighboring keyframes is lower than a threshold, (0.25 experimentally determined in this paper), then that keyframe was removed from the summary. Finally, keyframes with the standard deviation of pixel hue values less than 14 (possible black frames) are also removed from the summary.

3.1.1. Evaluation Metrics and Ground Truth

The ground truth summaries were collected through a user study conducted by De Avila *et al.* (2011), where 50 users participated, each one dealing with 5 videos, meaning that each video has 5 different user summaries, so totally 250 summaries were created manually, thus keeping the original opinion of every user. The quality of automatic summaries are assessed by the following two metrics (De Avila et al., 2011):

- Accuracy rate $CUS_A = n_{mAT}/n_{GT}$
- Error rate $CUS_E = n_{\bar{m}AT}/n_{GT}$

where n_{mAT} represents the number of matches between the automatic summary (AT) and ground truth user summary (GT), $n_{\bar{m}AT}$ represents the number of non-matched frames between AT and GT, and n_{GT} represents the number of frames in GT. From definition, it can be seen that the accuracy rate is smaller than or equal to one, however the error rate could be greater than one.

Table 1. The performance of different methods over the Open Video dataset.

Summarization Techniques	# selected KF	Avg # KF	CUS_A	CUS_E
DFP-ALC (our method)	452	434	**0.71**	**0.41**
RPCA-KFE (Dang and Radha, 2015)	383	434	0.64	**0.30**
STIMO (Furini et al., 2010)	496	434	**0.66**	0.62
DT (Mundur et al., 2006)	311	434	0.48	0.32

Fig. 4. Automatic video summary using the proposed DFP-ALC method over the video *Hurricane Force - A coastal perspective, segment 3* in the OV dataset.

Fig. 5. User summaries of the video *Hurricane Force - A coastal perspective, segment 3* in the OV dataset.

3.1.2. Quantitative Comparison

Here, we evaluate the keyframes only based on the similar image content without considering the timestamp between the keyframes inline with (Dang and Radha, 2015). This is because the OV videos are longer and the scenes tend to change slower. The evaluation is done in such a way that it is consistent with the human observer. The experimental results are presented in Table 1.

Therefore the following conclusions can be drawn for the OV dataset from Table 1: (i) The number of selected keyframes by our proposed DFP-ALC method (452 frames) is close to the average number of keyframes from the ground truth. It is not as small as 311 frames selected by the DT method or as large as 496 frames selected by the STIMO method; (ii) Since the number of the selected keyframes by our DFP-ALC method is higher than that of RPCA-KFE and lower than that of STIMO, the average error rate CUS_E, 0.41, of our method is lower than 0.62 of the STIMO method and the average accuracy rate CUS_A , 0.71, of our method is higher than 0.64 of the RPCA-KFE method. These results suggest that our proposed method

achieved better balance between accuracy and error rates than all the competitors selected.

3.1.3. Visual Comparison

Fig. 4 shows the automatic summary of our proposed method over the video *Hurricane Force - A coastal perspective, segment 3*. It contains 10 keyframes in which almost all the keyframes are in the ground truth user summaries (see Fig. 5) except the 4^{th} keyframe of the hurricane occurrence. Similarly, all the ground truth keyframes in Fig. 5 are in the retrieved automatic summary of our DFP-ALC method. These results show that while the participants had different opinions towards what frames should be selected as key ones, our automatic method can perform as well as an average participant.

3.1.4. Computational Efficiency

Since the source codes of the techniques compared were unavailable, we intend to present the time taken of our proposed method for all the 50 videos in the OV dataset. It took around 27.9 minutes in total. While the total duration of all the 50 videos is 74.49 minutes, our proposed method took nearly one-third of the actual duration of all the original videos, thus producing real-time video summaries.

3.1.5. Parameter Analysis

In this section, we experimentally investigate the impact of various parameters in our proposed method used for key frame extraction such as (i) frame size and patch size for distinct patch classification, (ii) entropy analysis for the computation of the DFP index, and (iii) color space for clustering.

Fig. 6. User summary # 1 of the video *Drift Ice as a Geologic Agent, segment 07* in the OV dataset.

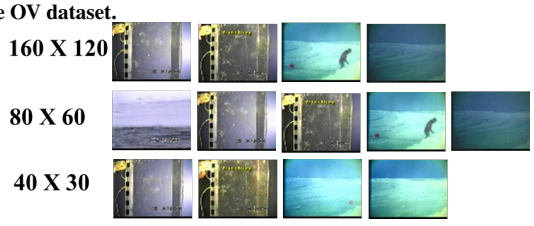

Fig. 7. The keyframes selected by the proposed method with the frames resized into different sizes for the video *Drift Ice as a Geologic Agent, segment 07* in the OV dataset.

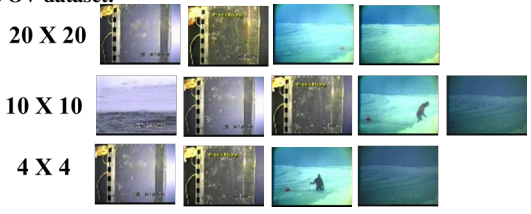

Fig. 8. The keyframes selected by the proposed method for different patch sizes of the video *Drift Ice as a Geologic Agent, segment 07* in the OV dataset.

We firstly evaluate the impact of different frame sizes and patch sizes using the video *Drift Ice as a Geologic Agent, segment 07* relative to its user summary #1 shown in Fig. 6. The keyframes retrieved by the proposed method with various frame sizes and patch sizes are shown in Figs. 7 and 8 respectively. It can be seen that the frame sizes of 160×120, 80×60 and 40×30 retrieved 3, 4 and 2 matched frames with a computational time of 210, 20 and 6 seconds, the patch sizes of 20×20, 10×10 and 4×4 retrieved 2, 4 and 3 matched frames with a computational time of 8, 20 and 203 seconds respectively. Such results show that our selection of the frame size of 80×60 and the patch size of 10×10 has achieved a good compromise between computational efficiency and keyframe retrieval accuracy.

We also evaluated the impact of two different entropy measures on our algorithm: (i) Shannon's entropy described in our proposed method (see Subsection 2.1.4) and (ii) Tsallis entropy (De Albuquerque et al., 2004). Our experimental results show that the proposed method is not sensitive to the selection of the entropy measures in terms of either the keyframe accuracy or computational time. Hence, Shannon's entropy is a sensible choice in our proposed method.

Fig. 9. User summary # 1 of the video *Take Pride in America, segment 01* in the OV dataset.

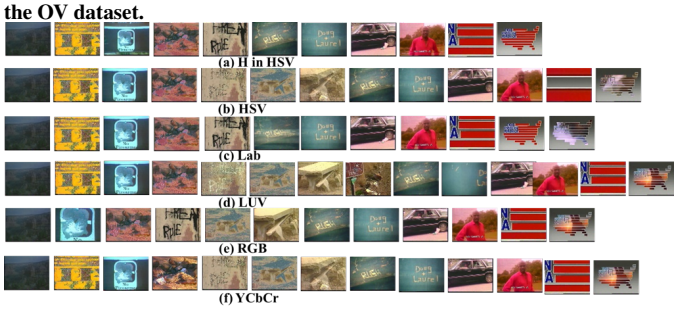

Fig. 10. The keyframes selected by the proposed method using different color spaces for clustering for the video *Take Pride in America, segment 01* in the OV dataset.

Finally, we analyse the impact of various color spaces for clustering the candidate frames using the video *Take Pride in America, segment 01* relative to its user summary # 1 shown in Fig. 9. We considered Hue alone in the HSV with 16 bins, and to approximate the total number of bins to 256 for all the other color spaces and also to give importance to color component, we took HSV with 16, 2 and 8 bins, RGB with 6, 7 and 6 bins for each corresponding color channel. In the RGB color space, we assigned 7 bins to the green channel, since it is close to the human perception of brightness. Each frame in the Lab, LUV and YCbCr color spaces was represented as a histogram with 4, 8, and 8 bins along different color channels as shown in Fig. 10, where it retrieved 9, 8, 9, 8, 8 and 8 matches with a computational time of 46, 60, 57, 61, 51 and 53 seconds respectively. Hence, we chose Hue alone in the HSV for clustering candidate frames in the OV dataset due to its increased accuracy and computational efficiency.

3.2. Consumer Video Dataset

While previous work were mostly applied to structured videos with certain publicly available datasets, we focus on consumer videos in this section, which is more challenging to summarize than structured professionally generated ones (e.g. news, documentary, sports, etc.). Our experiments were performed on 8 video clips from the Kodak Home Video Database (Luo et al., 2009). These clips were taken using KodakEasyShare C360 and V550 zoom digital cameras, with a VGA resolution of 640 × 480. The summary description of these clips is provided in Table 2. They vary in duration from 194 to 656 frames with 4 to 6 keyframes in the ground truth. We compare our proposed method with Heterogeneity image patch index (HIP) (Dang and Radha, 2014), Motion based keyframe extraction (MKFE) (Luo et al., 2009) and Equally Spaced Key Frames (ESKF). ESKF splits a video into n equal length segments and the last frame from each segment is selected, where n is the number of frames in the ground truth. In order to extract the candidate frames for clustering in our proposed method, we split the video into equal segments containing 10 frames each and a frame with the maximum DFP index in each segment was selected. The selected candidate frame t is represented as a color histogram C_t in the YCbCr color space with 16, 1 and 1 bin for luminance and chrominance respectively. The experimental results of different methods are presented in Table 3.

3.2.1. Quantitative Comparison

To find the matches between the automatically retrieved keyframes and those in the ground truth, we applied a two-way search followed by a consistency check method (Kannappan et al., 2016). The total number of reliable matches depicted in Table 3 indicates the pertinent matches obtained over all the 8 videos under each technique. The remaining keyframes could be considered as the non-matched frames or weak and false matches. Since the desired number of keyframes extracted by the automatic summaries of all the compared techniques are set equal to that in the ground truth, the performance metrics such as precision, recall and f-measure are all equal (precision = recall = f-measure). Table 3 indicates that our DFP-ALC method performs better for almost all the videos except one video *FireworkAndBoat*, when compared with other summarization techniques. Moreover, the number of reliable matches of our method increases thereby increasing the accuracy. This shows that our proposed *DFP-ALC* method is powerful in characterizing and retrieving the contents of a video.

3.2.2. Qualitative Comparison

It is interesting to consider the video *Skyline from overlook*, which contains 6 ground truth keyframes captured outdoors with significant amounts of change in perspective and brightness as shown in Fig. 11. From Fig. 12, it can be seen that ESKF delivers a strong baseline, even outperforming existing summarization technique, MKFE (Luo et al., 2009), since it retains temporal order and provides a pretty decent summary of the original video. However, this might not be the case for professional videos with slowly changing scenes. The proposed DFP-ALC method recalled the most matches from the ground truth. Even though all the four methods identify the keyframe with two persons standing on the seaside, their positions do vary from the right hand side to the left hand side. This reveals that it is challenging to identify the true matches between the automatic summary and the ground truth.

Fig. 11. User summary of the video *Skyline from overlook* in the consumer video dataset.

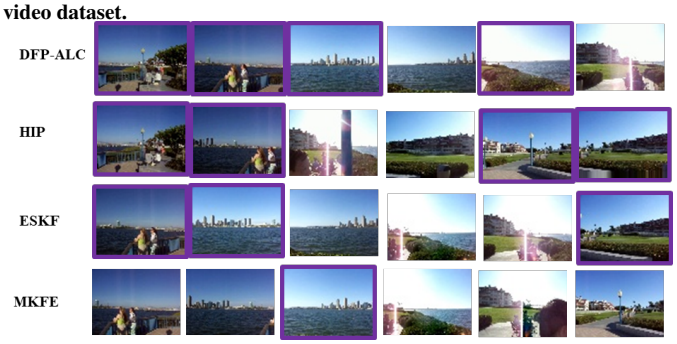

Fig. 12. Automatic video summaries of various summarization techniques of the video *Skyline from overlook* in the consumer video dataset. Purple bounding box indicates a pertinent match.

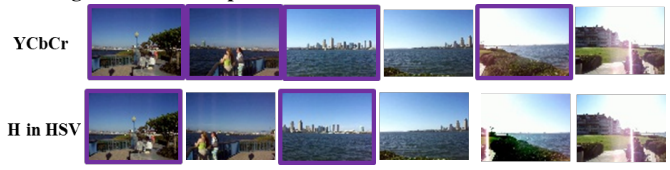

Fig. 13. The keyframes selected by the proposed method using YCbCr and Hue alone in the HSV color space for clustering for the video *Skyline from overlook* in the consumer video dataset. Purple bounding box indicates a pertinent match.

3.2.3. Computational Efficiency

Since the source code of the MKFE method was not available, we intend to compare the time complexity of our proposed method with the HIP method. Our proposed DFP-ALC method took only 48 seconds altogether to summarize all the 8 video clips in Table 3, whereas the HIP method took as much as 8.46 minutes. Therefore, the former is almost an order faster than the latter. On the other hand, the time taken for the former to summarize all the 8 video clips is just about half of their total duration (1.46 minutes), which clearly shows that our proposed method is able to produce real-time video summaries.

3.2.4. Ablation Study

While we use the same frame size, patch size and the entropy measure as in the OV dataset, the only distinction is the usage of the YCbCr color space with 16,1 and 1 bin for clustering the candidate frames. The selection of appropriate color space for representation, is based on some prior knowledge of the video content. To this end, since the scenes are very similar in the consumer video dataset and the illumination in the YCbCr color space help to retrieve distinct key frames, we chose it for clustering the candidate frames, where it retrieved more reliable matches as shown in Fig. 13 relative to its user summary shown

Table 2. The summary of Kodak video clips used for evaluation (Luo et al., 2009).

Video Name	# of keyframes (Ground Truth)	Total # of frames	Indoor/ Outdoor	Camera Motion	Persp. Changes	Bright. Changes
HappyDog	4	376	Outdoor	Yes	Yes	Yes
MusuemExhibit	4	250	Indoor	Yes	No	No
SoloSurfer	6	618	Outdoor	Yes	Yes	Yes
SkylinefromOverlook	6	559	Outdoor (dark)	Yes	Yes	Yes
FireworkAndBoat	4	656	Outdoor	Yes	No	No
BusTour	5	541	Outdoor (inside bus)	Yes	Yes	Yes
LiquidChocolate	6	397	Indoor	Yes	Yes	Yes
OrnateChurch	4	194	Outdoor	Yes	Yes	Yes

Table 3. The performance of different methods over the Kodak consumer video dataset.

Video Name	# of true matches				# of keyframes (Ground Truth)
	DFP-ALC (our method)	HIP	ESKF	MKFE	
HappyDog	**2**	2	2	2	4
MusuemExhibit	**3**	3	3	2	4
SoloSurfer	**3**	3	2	2	6
SkylinefromOverlook	**4**	4	3	1	6
FireworkAndBoat	2	2	2	3	4
BusTour	**3**	2	2	2	5
LiquidChocolate	**4**	4	4	3	6
OrnateChurch	**3**	3	2	3	4
Total # of reliable matches	**24**	23	20	18	39
Accuracy	**62%**	59 %	51 %	46 %	

in Fig. 11. It is interesting to note from Fig. 13 that, Hue alone in HSV with 16 bins also retrieve almost similar frames, except the 2^{nd} and the 5^{th} matched pair, which vary slightly in their panning position, which shows that the proposed approach is not much sensitive to feature definition across various datasets.

4. Conclusions

We proposed a novel keyframe extraction approach in order to produce static video summaries. It contains two main steps: (i) candidate frame selection using the proposed DFP index, and (ii) keyframe extraction using the proposed appearance based linear clustering of the candidate frames. We validated the proposed method over both the open video dataset and the consumer video dataset. The experimental results obtained are compared with the ground truth, selected by human judges, as well as with those selected by other state-of-the-art methods. These results show that the proposed DFP-ALC method outperforms the state-of-the-art ones in terms of both accuracy and computational efficiency. This is because the DFP index is powerful in describing the contents of the video frames for the selection of the candidate keyframes, and the appearance based linear clustering is effective in finding the representative ones. In the future, this work can be extended in the generation of video skims by combining the shots of the respective keyframes.

Acknowledgments

The first author is grateful for the award given by Aberystwyth University under the Departmental Overseas Scholarship (DOS) and partly funding by Object Matrix, Ltd on the project. We are also grateful to the anonymous reviewers for their insightful comments that have improved the clarity and the readability of the paper.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., Süsstrunk, S., 2012. Slic superpixels compared to state-of-the-art superpixel methods. TPAMI 34, 2274–2282.

Dang, C., Radha, H., 2015. Rpca-kfe: Key frame extraction for video using robust principal component analysis. TIP 24, 3742–3753.

Dang, C.T., Radha, H., 2014. Heterogeneity image patch index and its application to consumer video summarization. TIP 23, 2704–2718.

De Albuquerque, M.P., Esquef, I.A., Mello, A.G., 2004. Image thresholding using tsallis entropy. Pattern Recognition Letters 25, 1059–1065.

De Avila, S.E.F., Lopes, A.P.B., da Luz, A., de Albuquerque Araújo, A., 2011. Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters 32, 56–68.

Furini, M., Geraci, F., Montangero, M., Pellegrini, M., 2010. Stimo: Still and moving video storyboard for the web scenario. Multimedia Tools and Applications 46, 47–69.

Kanehira, A., Van Gool, L., Ushiku, Y., Harada, T., 2018. Viewpoint-aware video summarization, in: CVPR, pp. 7435–7444.

Kannappan, S., Liu, Y., Tiddeman, B., 2016. A pertinent evaluation of automatic video summary, in: ICPR, IEEE. pp. 2240–2245.

Luo, J., Papin, C., Costello, K., 2009. Towards extracting semantically meaningful key frames from personal video clips: from humans to computers. IEEE Transactions on Circuits and Systems for Video Technology 19, 289–301.

Mahasseni, B., Lam, M., Todorovic, S., 2017. Unsupervised video summarization with adversarial lstm networks, in: CVPR, pp. 202–211.

Mundur, P., Rao, Y., Yesha, Y., 2006. Keyframe-based video summarization using delaunay clustering. International Journal on Digital Libraries 6, 219–232.

Truong, B.T., Venkatesh, S., 2007. Video abstraction: A systematic review and classification. ACM Transactions on Multimedia Computing, Communications, and Applications 3, 3.

Vermaak, J., Pérez, P., Gangnet, M., Blake, A., 2002. Rapid summarisation and browsing of video sequences., in: BMVC, pp. 1–10.

Wu, J., Zhong, S.h., Jiang, J., Yang, Y., 2017. A novel clustering method for static video summarization. Multimedia Tools and Applications 76, 9625–9641.

Zhang, K., Chao, W.L., Sha, F., Grauman, K., 2016a. Summary transfer: Exemplar-based subset selection for video summarization, in: CVPR, pp. 1059–1067.

Zhang, K., Chao, W.L., Sha, F., Grauman, K., 2016b. Video summarization with long short-term memory, in: ECCV, Springer. pp. 766–782.

Zhang, K., Grauman, K., Sha, F., 2018. Retrospective encoders for video summarization, in: ECCV, pp. 383–399.

Zhuang, Y., Rui, Y., Huang, T.S., Mehrotra, S., 1998. Adaptive key frame extraction using unsupervised clustering, in: Proceedings of the International Conference on Image Processing, pp. 866–870.